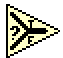
Toma de decisiones con la función Select

Los instrumentos virtuales descritos en este curso se ejecutan de manera ordenada siguiendo el flujo de datos. Pero hay casos en los que hay que tomar decisiones. Por ejemplo, si sucede **a**, hacer **b**; o si sucede **c**, hacer **d**.

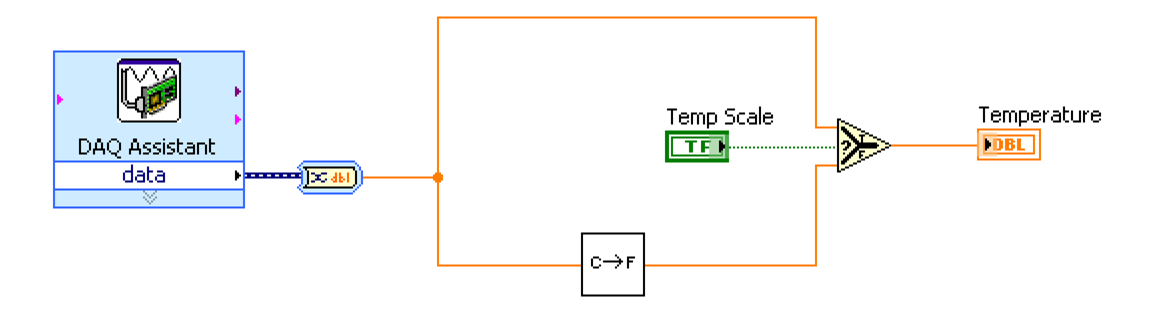
En programas basados en texto, esto se hace mediante sentencias if - else, switch - case, etc. LabVIEW incluye muchas y diferentes maneras de tomar decisiones. El método más simple de todos ellos es la función `Select`.

## Select Function



La función `Select`, situada en la paleta `Functions>>Express Comparison`, hace una selección entre dos valores según el valor que tome una entrada Booleana. Si la entrada Booleana`s` es `Cierta`, esta función devuelve el valor conectado a la entrada `t (true)`. Si la entrada Booleana es `Falsa`, la función devuelve el valor conectado a la entrada `f (false)`.

El ejercicio Thermometer VI utiliza la función `Select` para determinar si se desea la salida en grados Centígrados o en grados Fahrenheit, tal como se observa en el diagrama de bloques de la [link].



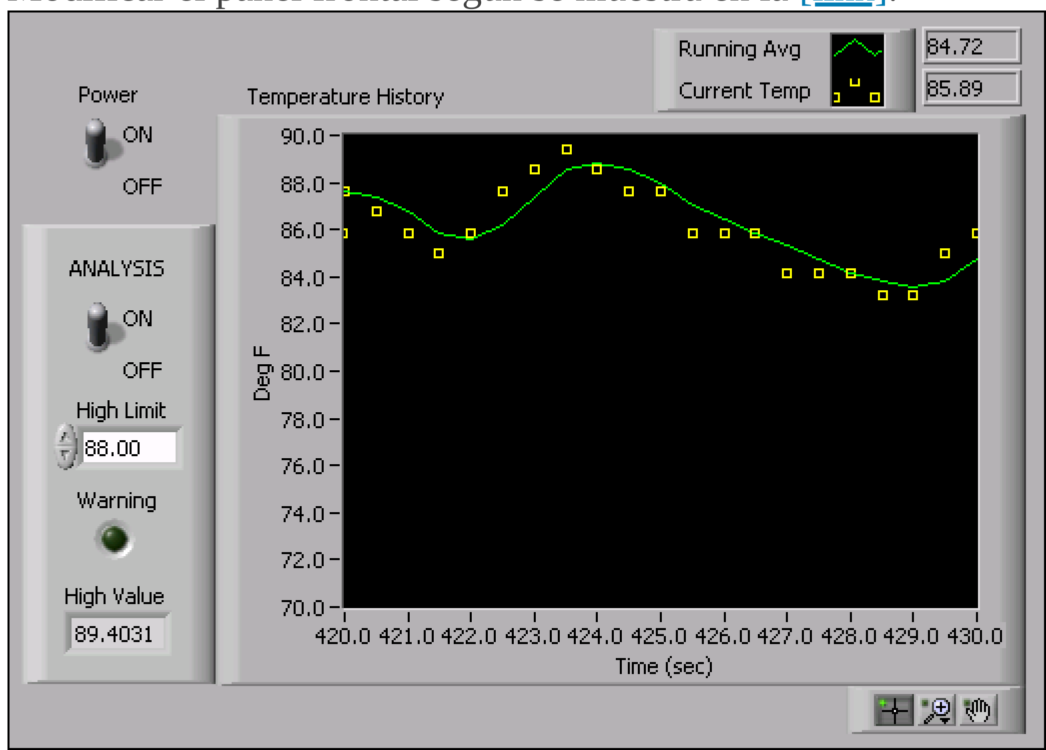

Si la decisión a tomar tiene una complejidad mayor que la que se puede tomar con la función `Select`, entonces se requerirá una estructura `Case`.

VI Control de Temperatura
**Exercise:**

**Problem:**

## Panel Frontal

1. Abrir el [VI Promedio de temperatura].
2. Modificar el panel frontal según se muestra en la [link].




1. Colocar en el panel frontal una caja con los bordes izquierdo y derecho suaves (horizontal smooth box), situada en la paleta `Controls>>All Controls>>Decorations`. Este objeto decorativo agrupa los elementos de `Analysis`.
2. Crear un duplicado del interruptor booleano `Power`. Etiquetar el nuevo interruptor como `ANALYSIS`. Hacer clic con el botón derecho sobre el interruptor y seleccionar en el menú desplegable `Mechanical Action>>Switch When Pressed`.
3. Colocar en el panel frontal un control numérico, situado en la paleta `Controls>>Numeric Controls`. Etiquetar el

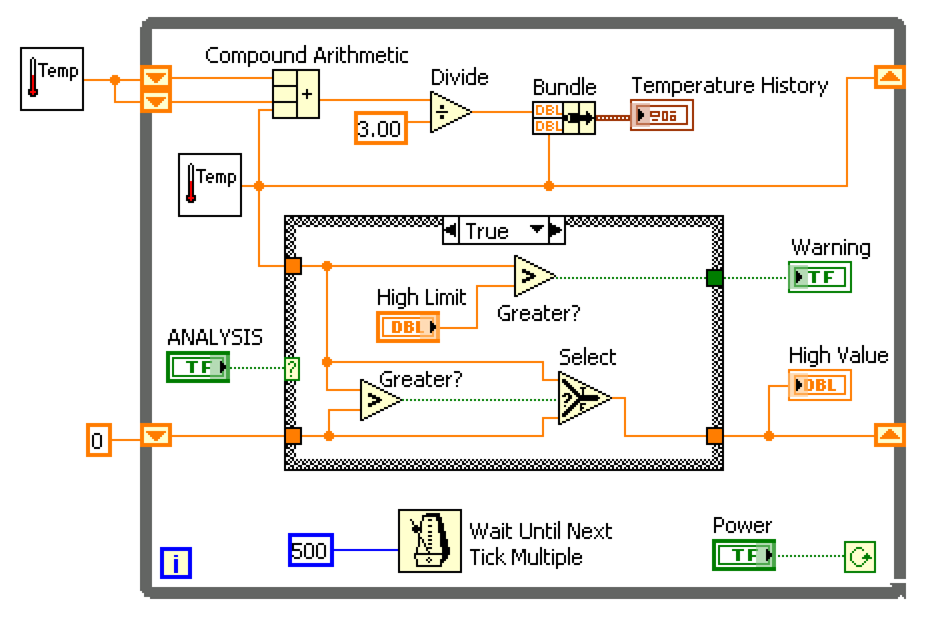
control `High Limit`.

4. Colocar un LED redondo en el panel frontal, situado en la paleta `Controls>>LEDs`. Etiquetar el indicador como `Warning`.
5. Colocar también en el panel frontal un indicador numérico situado en la paleta `Controls>>Numeric Indicators`. Etiquetar este indicador como `High Value`.
6. Hacer clic con el botón derecho sobre la pantalla del visualizador y seleccionar en el menú contextual `Visible Items>>Digital Display` para mostrar los valores digitales.

3. Seleccionar `File>>Save As` para salvar el instrumento virtual como `Control de Temperatura.vi` en el directorio `C:\Exercises\LabVIEW Basics I`.

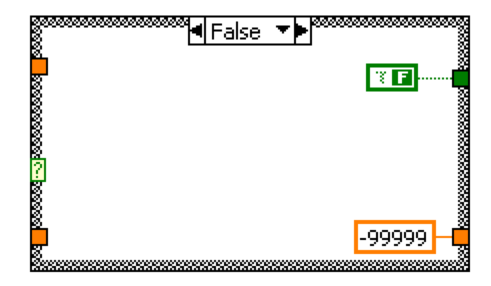
## Diagrama de Bloques

1. Modificar el Diagrama de Bloques como se muestra en la [link]. Agrandar el `While Loop` para crear espacio donde insertar los elementos nuevos.



1. Colocar en el diagrama de bloques la estructura `Case`, situada en la paleta `Functions>>Execution Control`. Conectar el control `Analysis` al terminal `selector` terminal. Hacer clic en el botón `decrement` o `increment` para elegir el case `True`.
2. Colocar en el diagrama de bloques dos funciones `Greater?`, situadas en la paleta `Functions>>Arithmetic & Comparison>>Express Comparison`. La primera función `Greater?` devuelve `True` si la temperatura rebasa `High Limit`. En cualquier otro caso, la función devuelve `False`.
3. Colocar en el diagrama de bloques la función `Select` situada en la paleta `Functions>>Arithmetic & Comparison>>Express Comparison`. Esta función devuelve el mayor de los dos valores.
4. Completar el caso `True` de la `Estructura Case` como se muestra en la [link].
5. Hacer clic en el botón `decrement` o `increment` para elegir el case `False`.

2. Completar el caso `False` de la `Estructura Case`, mostrado en la [link].




   1. Hacer clic con el botón derecho sobre el túnel que conecta con el `Warning Indicator` y seleccionar en el menú que aparece `Create>>Constant`. Usar la herramienta `Operating` para cambiar la constante a `False`.
   2. Hacer clic con el botón derecho sobre el túnel que conecta con el indicador `High Value` y seleccionar en el menú que aparece `Create>>Constant`. Escribe `-99999` como valor de la constante.

3. Salvar el VI. Este VI se volverá a usar más adelante a lo largo del curso.

## Ejecutar el VI

1. Mostrar el Panel Frontal, escribir `80` en `High Limit`, y ejecutar el VI. Si `ANALYSIS` está apagado, el VI apaga el LED `Warning` y muestra un valor de `-99999` en `High Value`. Si `ANALYSIS` está encendido, el VI enciende el LED `Warning` cuando la temperatura se encuentre por encima del valor `High Limit` y muestra el valor de temperatura actual `High Value`.
2. Cerrar el VI.

Formula Node

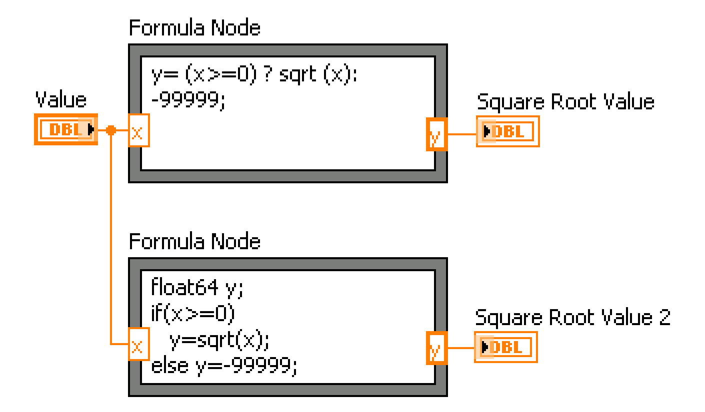
`Formula Node` se usa para realizar operaciones matemáticas en LabVIEW. Para obtener una funcionalidad añadida se puede enlazar con la aplicación matemática MATLAB®.

## Formula Node

`Formula Node` es un nodo basado en texto que se utiliza para realizar operaciones matemáticas en el diagrama de bloques. Los `Formula Node` son útiles para ecuaciones que tienen muchas variables o que sean complicadas de resolver o para usar código basado en texto ya existente. Es mucho más fácil copiar y pegar el código basado en texto en un `Formula Node` que representarlo gráficamente en el diagrama de bloques.

Para crear los terminales de entrada y salida de un `Formula Node` se puede hacer clic con el botón derecho sobre el borde del nodo y seleccionar en el menú contextual `Add Input` o `Add Output`, y después introducir la variable para la entrada o para la salida. Cada sentencia debe terminar con un punto y coma (`;`).

Los `Formula Node` también se pueden usar para llevar a cabo una toma de decisiones. El diagrama mostrado en la [link] muestra 2 maneras diferentes de usar una sentencia `if-then` en un `Formula Node`. Las 2 estructuras tendrán el mismo resultado.



El `Formula Node` puede ejecutar muchas operaciones diferentes. Consultar la ayuda de *LabVIEW* para una mayor información sobre las funciones, operaciones y sintaxis de los `Formula Node`.

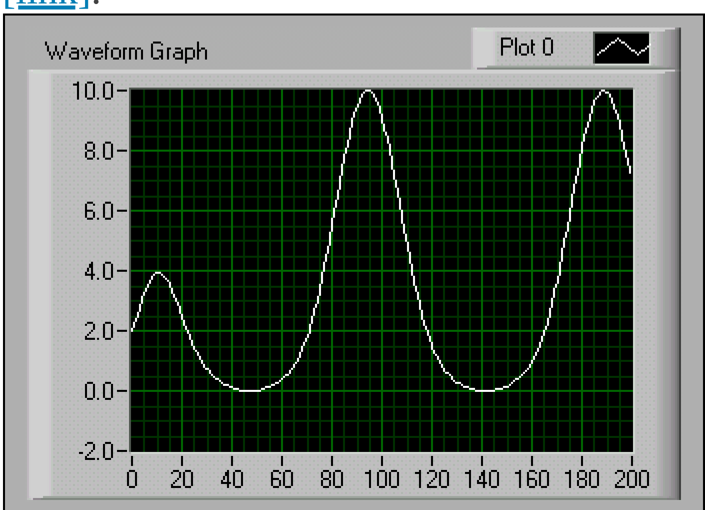
> **Note:** El VI Express
>
> `Formula`
>
> situado en la paleta
>
> `Functions>>Arithmetic & Comparison`
>
> usa un interface de calculadora para crear fórmulas matemáticas. Se puede usar este VI Express para ejecutar la mayoría de las operaciones que puede realizar una calculadora científica básica. Consultar la ayuda de *LabVIEW* para una mayor información sobre los VI Express
>
> `Formula`
>
> .

VI Ejercicio con Formula Node
**Exercise:**

**Problem:**

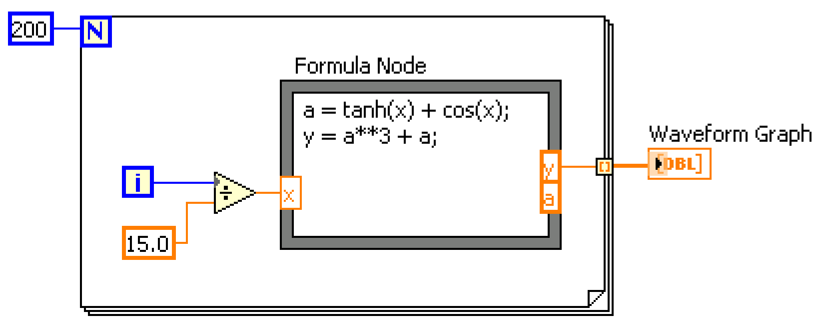
## Panel Frontal

1. Abrir un VI en blanco y construir el panel frontal mostrado en la [link].




## Diagrama de Bloques

1. Construir el diagrama de bloques mostrado en la [link].




1. Colocar en el diagrama de bloques el `Formula Node`, situado en la paleta `Functions>>All Functions>>Structures`.

2. Crear el terminal de entrada `x` haciendo clic con el botón derecho del ratón sobre el borde izquierdo y seleccionando `Add Input` en el menú contextual. Escribe `x` en la caja que aparece.
3. Crear los terminales de salida `y` y `a` haciendo clic con el botón derecho del ratón sobre el borde izquierdo y seleccionando `Add Output` en el menú contextual. Escribir `y` y `a` respectivamente, en las cajas que aparecen. Hay que crear terminales de salida para las variables temporales como $a$.

> **Note:** Cuando se crea un terminal de entrada o de salida, se tiene que usar para la variable el mismo nombre que el que se ha usado en la ecuación. Los nombres de variable son sensibles a mayúsculas y minúsculas.

4. Escribir las siguientes ecuaciones en el `Formula Node`, donde `**` es el operador exponencial. Consultar la ayuda de *LabVIEW* para una mayor información sobre la sintaxis de los `Formula Node`. `a = tanh(x) + cos(x); y = a**3 + a;`
5. Completar el diagrama de bloques según se muestra en la [link].

2. Salvar el VI como `Ejercicio Formula Node.vi` en el directorio `C:\Exercises\ LabVIEW Basics I`.

## Ejecutar el VI

1. Mostrar el panel frontal y ejecutar el VI. El gráfico muestra la representación de la ecuación $y = f^3(x) + f(x)$, donde $f(x) = \tanh(x) + \cos(x)$. Durante cada vuelta del bucle, el VI divide el terminal de contaje entre `15.0`. El cociente está unido

al `Formula Node`, que calcula el valor de la función. El instrumento virtual dibuja el array como un gráfico.

2. Cerrar el VI.

Resumen, trucos y pistas sobre la toma de decisiones en un VI

- La función `Select` permite elegir entre dos entradas en función del valor que tome una tercera entrada booleana.
- Una estructura `Case` tiene dos o más subdiagramas o casos. Solo se puede ver uno de los subdiagramas cada vez, y la estructura ejecuta solo un caso cada vez.
- Si el terminal selector del caso fuera una variable Booleana, la estructura tendría un caso `True` y un caso `False`. Si el terminal selector fuera un entero, una cadena, o un valor de tipo enumerado, la estructura entonces podría hasta $2^{31} - 1$ casos.
- Las entradas están disponibles para todos los subdiagramas de una estructura `Case`, pero los subdiagramas no tienen porque usar todas las entradas. Si al menos un túnel de salida queda sin definir, todos los túneles de salida en la estructura aparecerán como cuadros blancos.
- Cuando se crea un subVI a partir de una estructura `Case`, hay que conectar la entrada de `error` al terminal selector, y colocar en todos los subVI el código en el caso `No Error` para prevenir la ejecución del subVI si se recibe un error.
- Los `Formula Node` son útiles para ecuaciones que tienen muchas variables o que sean complicadas de implementar o para usar código basado en texto ya existente. Cada sentencia de la ecuación debe terminar con un punto y coma (`;`).

Más ejercicios sobre la toma de decisiones en un VI

**Exercise:**

**Problem:**

Construir un VI que haga uso de un `Formula Node` para calcular las siguientes ecuaciones:

$$y_1 = x^3 + x^2 + 5$$

$$y_2 = mx + b$$

Usar un único `Formula Node` para ambas ecuaciones y terminar cada ecuación con un punto y coma (`;`).

Salvar el VI como `Ecuaciones.vi` en el directorio `C:\Exercises\LabVIEW Basics I`.

**Exercise:**

**Problem:**

Construir un VI que funcione como una calculadora. En el panel frontal, usar controles numéricos para introducir dos números y un indicador numérico para mostrar el resultado de la operación (`Add`, `Subtract`, `Divide`, or `Multiply`) que el VI ejecute con dichos números. Usar un control deslizante para especificar la operación a realizar.

Salvar el VI como `Calculadora_4.vi` en el directorio `C:\Exercises\LabVIEW Basics I`.

**Exercise:**

**Problem:**

Modificar el Square Root VI para que ejecute todas las operaciones y el chequeo de la condición usando el `Formula Node`.

Seleccionar `File>>Save As` para salvar el VI como `Raiz Cuadrada 2.vi` en el directorio `C:\Exercises\LabVIEW Basics I`.

**Exercise:**

**Problem:**

Construir un VI con dos entradas, `Threshold` y `Input Array`, y una salida, `Output Array`. `Output Array` contiene los valores de `Input Array` que son mayores que el `Threshold`.

Salvar el VI como `Array sobre Umbral.vi` en el directorio `C:\Exercises\LabVIEW Basics I`.

Crear otro VI que genere un array de números aleatorios entre 0 y 1 y haga uso del `Array sobre Umbral VI` para obtener un array con valores mayores de 0.5.

Salvar el VI como `Usando Array sobre Umbral.vi` en el directorio `C:\Exercises\LabVIEW Basics I`.